Deep Learning (CSE641/ECE555)
Quiz-1 (15 Marks) (Duration 60 min)

Name .....
Roll No. .....

Beginner: 6 Marks

1. How regression and density estimation models can be used for classification? .5 Marks

   Regression uses class scores for classification, and Density estimation uses Baye's law.

2. Assume that the input X to some scalar function $f(.)$ is $n \times m$ matrix. What is the dimensionality of the gradient of f with respect to X? .5 Marks

   Same as X $n \times m$, because each element of this matrix represents the partial derivative of f with respect to the corresponding element of X.

3. Which one is more powerful - a two layer neural network without any activation function or a two layer binary decision tree & Why? .5 Marks

   NN. In BDT the decision nodes for depth 2 is 4, and the number of leaf nodes is 8.

4. What are the different types of learning methods in ML/DL? 1 Marks

   Lecture 1, slide 11

5. Your binary classification network is $y = \sigma(ReLU(z))$, where the predicted label of input is chosen to be 1 when $\hat{y} \geq 0.5$ and 0 otherwise. What will happen to this network while training? 1 Marks

   All samples will be labelled positive.

6. What happens to the receptive field of a 1D convolutional network as more layers are added? .5 Marks

   (a) It stays the same
   (b) It decreases exponentially
   (c) It increases linearly
   (d) It increases non-linearly

7. Which of the following is true about dropout? .5 Marks

   (a) Dropout leads to sparsity in the trained weights
   (b) At test time, dropout is applied with inverted keep probability
   (c) The larger the keep probability of a layer, the stronger the regularization of the weights in that layer
   (d) None of the above

8. Can we remove the bias parameter from the fully-connected layer and the convolutional layer before the batch normalization? 1 Marks

   Mathematically Yes, since the mean subtraction step in BN will cancel the bias and BN itself has a bias parameter. However, if BN is applied after the activation function, then not always.

9. According to the Universal Approximation Theorem, which type of function can a sufficiently wide feedforward neural network approximate on a compact subset of $\mathbb{R}^n$ .5 Marks

   (a) Only smooth functions
   (b) Only polynomial functions
   (c) Any continuous function
   (d) Only functions with bounded derivatives

Intermediate: 5 Marks

1. Which of these methods will use less CPU RAM: (a) loading model from individual weight files or (b) sequential model loading with meta device? .5 Mark

   individual weight files

2. What are the two broad ways to reduce/avoid overfitting (Hint: Think about function approx. in ERM framework)? .5 Marks

   Reducing the space (less functionals, simple DNN with fewer modules/layers); Making the choice of f^* less dependent on data (penalty on coefficients, margin maximization, ensemble methods)

3. Which of the following would you consider to be valid activation functions (elementwise non-linearities) to train a neural net in practice and Why? .5 Mark

   (a) $\phi(x) = -\min(2, x)$

   (b) $\phi(x) = 0.9x^2 + 1$

   (c) $\phi(x) = \begin{cases} \min(x, 0.1x) | x \geq 0 \\ \min(x, 0.1x) | x < 0 \end{cases}$

   (d) $\phi(x) = \begin{cases} \max(x, 0.1x) | x \geq 0 \\ \min(x, 0.1x) | x < 0 \end{cases}$

   (i), (ii), (iii). (iv) is linear functions, therefore quite useless as activations.

4. You are training a deep MLP (100 layers) on a binary classification task, using a sigmoid activation in the final layer and a mixture of tanh and ReLU activations for all other layers. You notice weights to a subset of layers stop updating after the first epoch of training, even though your network has not yet converged. Why is this happening? and explain which among the below options might help. 2 Marks

   (a) Increase the size of your training set

   (b) Switch the ReLU activations with leaky ReLUs everywhere

   (c) Add Batch Normalization before every activation

   (d) Increase the learning rate

   (ii), (iii).
   Classic vanishing gradient problem. Increasing size of the training set (i) doesn't help as the issue lies with the learning dynamics of the network. Varying the learning rate (iv) might help the network learn faster, but as the problem states the gradients to specific layers almost completely go to zero, so the issue seems to be localized to specific layers. (ii) Solves the problem of dying relus by passing some gradient signal back through all relu layers. (iii) Adding BatchNorm prior to every activation ensures the tanh layers have inputs distributed closer to the linear region of the activation, so the elementwise derivative across the layer evaluates closer to 1.

5. An input of size $[2 \times 3 \times 256 \times 256]$ passed through a 2D convolution layer with output channel (12), kernel size (5,5), dilation (2,2), padding=valid, stride (1,3). Find output shape. 1.5 Marks

   (2x12×248×83) Look formula of Height and Width of an image on PyTorch's 2D Convolution Documentation.

Advanced: 4 Marks

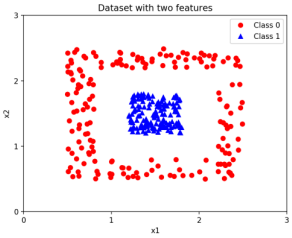
1. After the first DL assignment, your friend Ram concludes that Dropout and Batch Normalization (BN) often lead to a worse performance when they are combined together for CNNs (including ResNets). But your friend Sita argues against it by showing a counter-example of Wide-ResNet (WRN). Is Ram's conclusion wrong? Explain.

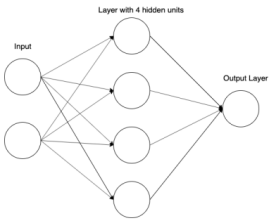
   Dropout shifts the variance of a specific neural unit when we transfer the state of that network from training to test. However, BN maintains its statistical variance, which is accumulated from the entire learning procedure, in the test phase. The inconsistency of variances in Dropout and BN i.e., the "variance shift") causes the unstable numerical behavior in inference that leads to erroneous predictions finally. Meanwhile, the large feature dimension in WRN further reduces the "variance shift" to bring benefits to the overall performance.

2. You have a dataset where each example contains two features, x1 and x2, and a binary label as shown below. You want to develop a model to perform binary classification using a single hidden layer with 4 neurons. If you use the below mentioned activation function is it possible for this model to achieve perfect accuracy on this dataset? If so, provide a set of weights that achieves perfect accuracy. If not, briefly explain why.

$$\phi(x) = \begin{cases} x \geq 0 : \frac{2x^2}{x+|x|} \\ x \leq 0 : 0 \end{cases}$$

   The key is to have each hidden node evaluate one of the sides of the separating square, and the output layer checks that all conditions are true (or false, depending on how the hidden weights are set). For example: w1 = (0, -1), b1 = 2; w2 = (-1, 0), b2 = 2; w3 = (0, 1), b3 = -1; w4 = (1, 0), b4 = -1; woutput = (1, 1, 1, 1), b1 = -4





Bonus Questions.

1. Derive an expression for the gradient of cross-entropy loss. 1 Point

   the cross-entropy loss function for a single example:

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

   Where:

   - y is the true label.
   - $\hat{y}$ is the predicted logits before applying softmax.
   - C is the number of classes.

   Apply chainrule

$$\frac{\partial L}{\partial \hat{y}_i} = \frac{\partial L}{\partial o_i} \frac{\partial o_i}{\partial \hat{y}_i}$$

   o_i is the raw output (logit) of the i-th class.

$$\frac{\partial L}{\partial o_i} = -\frac{\partial}{\partial o_i} \sum_{j=1}^{C} y_j \log(\hat{y}_j)$$

$$= -\frac{y_i}{\hat{y}_i}$$

$$\frac{\partial o_i}{\partial \hat{y}_i} = \hat{y}_i(1 - \hat{y}_i)$$

$$\frac{\partial L}{\partial \hat{y}_i} = -\frac{y_i}{\hat{y}_i} \hat{y}_i(1 - \hat{y}_i)$$

$$= -y_i(1 - \hat{y}_i)$$

   final gradient w.r.t logits

$$\frac{\partial L}{\partial \hat{y}_i} = \hat{y}_i - y_i$$

Deep Learning (CSE641/ECE555)
Quiz-1 (15 Marks) (Duration 60 min)

Name .....
Roll No. .....

Beginner: 6 Marks

1. How regression and density estimation models can be used for classification? .5 Marks

2. Assume that the input X to some scalar function $f(.)$ is $n \times m$ matrix. What is the dimensionality of the gradient of f with respect to X? .5 Marks

   Same as X $n \times m$, because each element of this matrix represents the partial derivative of f with respect to the corresponding element of X.

3. Which one is more powerful - a two layer neural network without any activation function or a two layer binary decision tree? Why? .5 Marks

4. What are the different types of learning methods in ML/DL? .5 Marks

5. Your binary classification network is $y = \sigma(ReLU(z))$, where the predicted label of an input is chosen to be 1 when $\hat{y} \geq 0.5$ and 0 otherwise. What will happen to this network while training? 1 Marks

6. Name three types of functions for which UAT doesn't hold true. 1 Marks

   functions that are non-continuous or defined over an open interval or over an infinitely wide domain.

7. Which of the following is true about dropout? .5 Marks

   (a) Dropout leads to sparsity in the trained weights

   (b) At test time, dropout is applied with inverted keep probability

   (c) The larger the keep probability of a layer, the stronger the regularization of the weights in that layer

   (d) None of the above

8. Can we remove the bias parameter from the fully-connected layer and the convolutional layer before the batch normalization? 1 Marks

9. The maximal number of linear regions of functions computed by a single layer rectifier network with n_0 inputs and n_1 hidden units is? .5 Marks

   $\sum_{j=0}^{n_0} \binom{n_1}{j}$

Intermediate: 5 Marks

1. Which of the following would you consider to be valid activation functions (elementwise non-linearities) to train a neural net in practice and Why? .5 Mark

   (a) $\phi(x) = -\min(2, x)$

   (b) $\phi(x) = 0.9x^2 + 1$

   (c) $\phi(x) = \begin{cases} \min(x, 0.1x) | x \geq 0 \\ \min(x, 0.1x) | x < 0 \end{cases}$

   (d) $\phi(x) = \begin{cases} \max(x, 0.1x) | x \geq 0 \\ \min(x, 0.1x) | x < 0 \end{cases}$

2. Name and briefly explain the three types of double decent phenomena in DL. .5 Marks

   Model-wise: Classical double descent; Sample-wise: There is a regime where more data hurts; Epoch-wise: Also called as Grokking (very popular in LLMs).

3. Consider the estimate MAP=$f_{Y|X}(y|x)f_X(x)$. Let X be a continuous random variable with the following PDF $f_X(x) = [2x\ (0 < x < 1); 0 \text{ otherwise}]$. Also suppose

$$P_{Y|X}(y|x) = x(1-x)^{y-1}, \quad \text{for } y = 1, 2, \cdots \ (\textit{Geometric dist.})$$

   Find the MAP estimate of X given $Y = 3$ .5 Marks

   $P_{Y|X}(y|x) = x(1-x)^{y-1} = P_{Y|X}(3|x) = x(1-x)^2$. Hence maximum of $P_{Y|X}(y|x)f_X(x) = x(1-x)^2 2x$ is $x = .5$

4. After training the model architecture with cross-entropy loss, you find that the softmax classifier works well. Specifically, the model achieves 100% accuracy on the training data. However, you observe that the training loss doesn't quite reached zero. How can you fix this? If not why? 2 Marks

   Given correct class 'c', the loss reduces to the term $\log \sum_i e^z - \log e^{z_c}$; as all the exponentials are non-zero, this loss cannot be zero, unless output probability=1, which is not achievable in finite computation.

5. An input of size $[4 \times 3 \times 256 \times 256]$ passed through a 2D convolution layer with output channel (12), kernel size (5,3), dilation (2,1), padding=valid, stride (1,3). Find output shape. 1.5 Marks

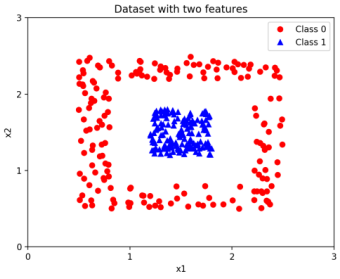
   (4x12×248×85) Look formulla of Height and Width of image on PyTorch's 2D Convolution Documentation.

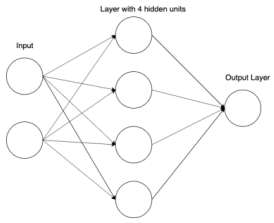
Advanced: 4 Marks

1. After the first DL assignment, your friend Ram concludes that Dropout and Batch Normalization (BN) often lead to a worse performance when they are combined together for CNNs (including ResNets). But your friend Sita argues against it by showing a counter-example of Wide-ResNet (WRN). Is Ram's conclusion wrong? Explain Yes/No.

2. You have a dataset where each example contains two features, x1 and x2, and a binary label as shown below. You want to develop a model to perform binary classification using a single hidden layer with 4 neurons. If you use the below mentioned activation function is it possible for this model to achieve perfect accuracy on this dataset? If so, provide a set of weights that achieves perfect accuracy. If not, briefly explain why.

$$\phi(x) = \begin{cases} x \geq 0 : \frac{2x^2}{x+|x|} \\ x \leq 0 : 0 \end{cases}$$





Bonus Question

1. Prove the following lower bound on the cross-entropy loss for an example considering K classes, softmax activation with cross-entropy loss and ground truth vector y as one-hot encoding. 1 Point

$$L_{CE}(\hat{y}, y) \geq K \log K$$

$$L_{CE}(\hat{y}, y) = -\sum \log \hat{y}_i \geq -K \log \sum \frac{\hat{y}}{K} (\text{Jensen's inequality}) = (-K) \log \frac{1}{K} (\text{softmax sums to 1}) = K \log K$$